UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/875,977	06/08/2001	Robert Vincent Michel Oerlemans	209694US2	8369

22850 7590 08/29/2005

OBLON, SPIVAK, MCCLELLAND, MAIER & NEUSTADT, P.C.
1940 DUKE STREET
ALEXANDRIA, VA 22314

EXAMINER	
ELMORE, JOHN E	

ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 08/29/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/875,977	OERLEMANS, ROBERT VINCENT MICHEL	
	Examiner	Art Unit	
	John Elmore	2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 June 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☒ Claim(s) *13-15* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 May 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-15 have been examined.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

3. **Claims 1-7 are rejected under 35 U.S.C. 102(e)** as being anticipated by Chrisop et al., hereafter Chrisop, (US Patent Number 09875977).

Regarding claim 1, Chrisop teaches a method of making information contents of memory cells of a volatile semiconductor memory irretrievable, said method comprising of:

a first step generating a digital pattern (paragraph 37, lines 14-18, and paragraph 44, lines 11-17) and

a second step of overwriting said information contents with said digital pattern at least two times (paragraph 44, lines 17-19).

Regarding claim 2, Chrisop teaches all the limitations of claim 1, and further teaches that said digital pattern overwrites said information contents alternately with its complementary pattern (e.g. writing a bit mask of all 1s followed by all 0s; paragraph 44, lines 11-27).

Regarding claim 3, Chrisop teaches all the limitations of claim 1, and further teaches that said digital pattern is a predefined digital pattern comprising both zeros and ones (paragraph 44, lines 11-17)

Regarding claim 4, Chrisop teaches all the limitations of claim 1, and further teaches that said information contents alternately with its a ratio of the number of zeros and the number of one in said predefined digital pattern is about one (checkerboard pattern; paragraph 44, line 26).

Regarding claims 5 and 6, Chrisop teaches all the limitations of claim 4, and further teaches that said ratio differs less than thirty percent from one, and that said ratio is one, respectively (checkerboard pattern exhibits ratio of one; paragraph 44, line 26).

Regarding claim 7, Chrisop teaches all the limitations of claim 1, and further teaches said digital pattern is a random pattern (paragraph 44, line 13).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. **Claims 8-12 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Kleijne (US Patent Number 4,593,384) in view of Chrisop.

Regarding claim 8, Kleinje teaches a device comprising:

a cryptographic chip (column 7, lines 17-24, and column 8, lines 56-58)

and a tampering signal generating device for generating a tampering signal (column 9, lines 7-11),

said cryptographic chip comprising

a volatile semiconductor memory having a plurality of memory cells (column 7, lines 31-35),

a control device for placing a cryptographic key in memory cells of said volatile semiconductor memory (column 7, lines 30-42),

a pattern generating device for generating a digital pattern (column 7, line 36),

an address generating device for generating addresses in said memory cells (data processor; column 7, lines 25-37),

said pattern generating device and said address generating device being connectable to said volatile semiconductor memory (column 7, lines 38-51),

said tampering signal generating device being connected to said pattern generating device and said address generating device (column 9, lines 7-17 and 51-63).

But Kleijne does not explain said pattern generating device and said address generating device being adapted for in response to a said tampering signal being connected to said volatile semiconductor memory and overwriting contents of said memory cells with a pattern based upon said digital pattern for at least two times.

However, Chrisop teaches overwriting contents of memory cells with a digital pattern for at least two times (paragraph 44, lines 17-19) in order to more securely erase the memory (paragraph 29). Therefore, it would be obvious to a person of ordinary skill in the computer art at the time the invention was made to modify the device of Kleijne with the teaching of Chrisop to provide a device wherein said pattern generating device and said address generating device are adapted for in response to a said tampering signal being connected to said volatile semiconductor memory and overwrite contents of said memory cells with a pattern based upon said digital pattern for at least two times. One would be motivated to do so in order to more securely erase the memory of the device.

Regarding claim 9, Kleijne and Chrisop teach all the limitations of claim 8, and further teach a device in which said cryptographic chip comprises

first connecting means connecting an output of said pattern generating device to a data input of said volatile semiconductor memory (Kleijne, column 7, lines 38-51, and Figure 9),

second connecting means for connecting said address generating device to an address input of said volatile semiconductor memory (Kleijne, column 7, lines 38-51, and Figure 9), and

a clock generator for generating clock signals for said pattern generating device and said address generating device (Kleijne, column 7, lines 29 and 38-51, and Figure 9).

Regarding claim 10, Kleijne and Chrisop teach all the limitations of claim 8, but Kleinje does not explain that said digital pattern is a predefined digital pattern comprising both zeros and ones.

However, Chrisop teaches overwriting contents of memory cells with a predefined digital pattern comprising both zeros and ones (paragraph 44, lines 11-17) in order to more securely erase the memory (paragraph 29). Therefore, it would be obvious to a person of ordinary skill in the computer art at the time the invention was made to modify the device of Kleijne with the teaching of Chrisop to provide a device wherein said digital pattern is a predefined digital pattern comprising both zeros and ones. One would be motivated to do so in order to more securely erase the memory of the device.

Regarding claim 11, Kleijne and Chrisop teach all the limitations of claim 8, but Kleinje does not explain that said digital pattern alternately is said digital pattern and a complementary pattern of said digital pattern.

However, Chrisop teaches overwriting contents of memory cells with a digital pattern that alternately is the digital pattern and a complementary pattern of the digital

pattern (paragraph 44, lines 11-47) in order to more securely erase the memory (paragraph 29). Therefore, it would be obvious to a person of ordinary skill in the computer art at the time the invention was made to modify the device of Kleijne with the teaching of Chrisop to provide a device wherein said digital pattern alternately is said digital pattern and a complementary pattern of said digital pattern. One would be motivated to do so in order to more securely erase the memory of the device.

6. **Claims 8-15 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Kleijne in view of Chrisop, as applied to claim 9, and further in view of Dallas Semiconductor Corp., hereafter Dallas, ("DS5002FP: Secure Microprocessor Chip," User Manual, 1995).

Regarding claim 12, Kleijne and Chrisop teach all the limitations of claim 9, and further teach a device with said device being adapted to have a power down state in which no main power is supplied to said device (Kleinje, low voltage detector, column 12, lines 5-9, and Figure 9). But the modified device of Kleijne and Chrisop does not explain that said device comprises a battery back up power supply, said pattern generating device, said clock generator and said address generating device being permanently connected to said back up battery power supply.

However, Dallas teaches a cryptographic chip comprising a battery back up power supply (lithium battery; page 61, column 1, lines 1-3) wherein a pattern generating device, a clock generator, and an address generating device are permanently connected to said back up battery power supply (the backup battery

powers the DC5002FP chip and all devices connected to it; page 61, column 2, lines 1-6) in order to provide a more secure device better able to resist tampering from an array of threats (page 1, column 1, lines 1-11). Therefore, it would be obvious to a person of ordinary skill in the computer art at the time the invention was made to modify the device of Kleijne and Chrisop with the teaching of Dallas to provide a device comprising a battery back up power supply, said pattern generating device, wherein said clock generator and said address generating device being permanently connected to said back up battery power supply. One would be motivated to do so in order to provide a more secure device better able to resist tampering from an array of threats.

Allowable Subject Matter

7. **Claims 13-15 are objected to as being dependent upon a rejected base claim**, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject matter:

The closest prior art, the modified device of Kleijne, Chrisop, and Dallas, does not explain a second rate for clocking said address generating device at a rate greater than a first rate (claim 13), which would include a second rate substantially greater than a first rate (claim 14) as well as a second rate that would permit memory cells to be addressed at least three times within 1 millisecond (claim 15). While it was generally known to persons of ordinary skill in the computer art at the time the invention was

made to utilize multiple clock rates within semiconductor devices, the prior art only discloses that the pattern generating device and the address generating device used to erase the memory cells operate at the same clock rate as the host device; the particular use of a second rate to control memory erasure via an anti-tampering mechanism is not publicly disclosed and is a non-obvious modification over the prior art.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Rasmussen et al. (US Patent Number 5,237,611) discloses a cryptographic device that erases its volatile memory upon tampering.

Siebold et al. (US Patent Number 4,720,700) discloses a security device for a car radio that, upon tampering, erases a cryptographic code in its memory that is required for operation of the radio.

Kaule (US Patent Number 4,783,801) discloses a device for protecting secret information which uses a battery backup mechanism to erase memory contents upon detection of tampering.

Gutmann, P., "Secure Deletion of Data from Magnetic and Solid-State Memory," Sixth USENIX Security Symposium Proceedings, July 1996.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John Elmore whose telephone number is 571-272-4224. The examiner can normally be reached on M 10-8, T-Th 9-7.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Greg Morse can be reached on 571-272-3838. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JE

GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Secure Deletion of Data from Magnetic and Solid-State Memory

Peter Gutmann
Department of Computer Science
University of Auckland
pgut001@cs.auckland.ac.nz

This paper was first published in the Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996

Abstract

With the use of increasingly sophisticated encryption systems, an attacker wishing to gain access to sensitive data is forced to look elsewhere for information. One avenue of attack is the recovery of supposedly erased data from magnetic media or random-access memory. This paper covers some of the methods available to recover erased data and presents schemes to make this recovery significantly more difficult.

1. Introduction

Much research has gone into the design of highly secure encryption systems intended to protect sensitive information. However work on methods of securing (or at least safely deleting) the original plaintext form of the encrypted data against sophisticated new analysis techniques seems difficult to find. In the 1980's some work was done on the recovery of erased data from magnetic media [1] [2] [3], but to date the main source of information is government standards covering the destruction of data. There are two main problems with these official guidelines for sanitizing media. The first is that they are often somewhat old and may predate newer techniques for both recording data on the media and for recovering the recorded data. For example most of the current guidelines on sanitizing magnetic media predate the early-90's jump in recording densities, the adoption of sophisticated channel coding techniques such as PRML, the use of magnetic force microscopy for the analysis of magnetic media, and recent studies of certain properties of magnetic media recording such as the behaviour of erase bands. The second problem with official data destruction standards is that the information in them may be partially inaccurate in an attempt to fool opposing intelligence agencies (which is probably why a great many guidelines on sanitizing media are classified). By deliberately under-stating the requirements for media sanitization in publicly-available guides, intelligence agencies can preserve their information-gathering capabilities while at the same time protecting their own data using classified techniques.

This paper represents an attempt to analyse the problems inherent in trying to erase data from magnetic disk media and random-access memory without access to specialised equipment, and suggests methods for ensuring that the recovery of data from these media can be made as difficult as possible for an attacker.

2. Methods of Recovery for Data stored on Magnetic Media

Magnetic force microscopy (MFM) is a recent technique for imaging magnetization patterns with high resolution and minimal sample preparation. The technique is derived from scanning probe microscopy (SPM) and uses a sharp magnetic tip attached to a flexible cantilever placed close to the surface to be analysed, where it interacts with the stray field emanating from the sample. An image of the field at the surface is formed by moving the tip across the surface and measuring the force (or force gradient) as a function of position. The strength of the interaction is measured by monitoring the position of the cantilever using an optical interferometer or tunnelling sensor.

Magnetic force scanning tunneling microscopy (STM) is a more recent variant of this technique which uses a probe tip typically made by plating pure nickel onto a prepatterned surface, peeling the resulting thin film from the substrate it was plated onto and plating it with a thin layer of gold to minimise corrosion, and mounting it in a probe where it is placed at some small bias potential (typically a few tenths of a nanoamp at a few volts DC) so that electrons from the surface under test can tunnel across the gap to the probe tip (or vice versa). The probe is scanned across the surface to be analysed as a feedback system continuously adjusts the vertical position to maintain a constant current. The image is then generated in the same way as for MFM [4] [5]. Other techniques which have been used in the past to analyse magnetic media are the use of ferrofluid in combination with optical microscopes (which, with gigabit/square inch recording density is no longer feasible as the magnetic features are smaller than the wavelength of visible light) and a number of exotic techniques which require significant sample preparation and expensive equipment. In comparison, MFM can be performed through the protective overcoat applied to magnetic media, requires little or no sample preparation, and can produce results in a very short time.

Even for a relatively inexperienced user the time to start getting images of the data on a drive platter is about 5 minutes. To start getting useful images of a particular track requires more than a passing knowledge of disk formats, but these are well-documented, and once the correct location on the platter is found a single image would take approximately 2-10 minutes depending on the skill of the operator and the resolution required. With one of the more expensive MFM's it is possible to automate a collection sequence and theoretically possible to collect an image of the entire disk by changing the MFM controller software.

There are, from manufacturers sales figures, several thousand SPM's in use in the

field today, some of which have special features for analysing disk drive platters, such as the vacuum chucks for standard disk drive platters along with specialised modes of operation for magnetic media analysis. These SPM's can be used with sophisticated programmable controllers and analysis software to allow automation of the data recovery process. If commercially-available SPM's are considered too expensive, it is possible to build a reasonably capable SPM for about US$1400, using a PC as a controller [6].

Faced with techniques such as MFM, truly deleting data from magnetic media is very difficult. The problem lies in the fact that when data is written to the medium, the write head sets the polarity of most, but not all, of the magnetic domains. This is partially due to the inability of the writing device to write in exactly the same location each time, and partially due to the variations in media sensitivity and field strength over time and among devices.

In conventional terms, when a one is written to disk the media records a one, and when a zero is written the media records a zero. However the actual effect is closer to obtaining a 0.95 when a zero is overwritten with a one, and a 1.05 when a one is overwritten with a one. Normal disk circuitry is set up so that both these values are read as ones, but using specialised circuitry it is possible to work out what previous "layers" contained. The recovery of at least one or two layers of overwritten data isn't too hard to perform by reading the signal from the analog head electronics with a high-quality digital sampling oscilloscope, downloading the sampled waveform to a PC, and analysing it in software to recover the previously recorded signal. What the software does is generate an "ideal" read signal and subtract it from what was actually read, leaving as the difference the remnant of the previous signal. Since the analog circuitry in a commercial hard drive is nowhere near the quality of the circuitry in the oscilloscope used to sample the signal, the ability exists to recover a lot of extra information which isn't exploited by the hard drive electronics (although with newer channel coding techniques such as PRML (explained further on) which require extensive amounts of signal processing, the use of simple tools such as an oscilloscope to directly recover the data is no longer possible).

Using MFM, we can go even further than this. During normal readback, a conventional head averages the signal over the track, and any remnant magnetization at the track edges simply contributes a small percentage of noise to the total signal. The sampling region is too broad to distinctly detect the remnant magnetization at the track edges, so that the overwritten data which is still present beside the new data cannot be recovered without the use of specialised techniques such as MFM or STM (in fact one of the "official" uses of MFM or STM is to evaluate the effectiveness of disk drive servo-positioning mechanisms) [7]. Most drives are capable of microstepping the heads for internal diagnostic and error recovery purposes (typical error recovery strategies consist of rereading tracks with slightly changed data threshold and window offsets and varying the head positioning by a few percent to either side of the track), but writing to the

media while the head is off-track in order to erase the remnant signal carries too much risk of making neighbouring tracks unreadable to be useful (for this reason the microstepping capability is made very difficult to access by external means).

These specialised techniques also allow data to be recovered from magnetic media long after the read/write head of the drive is incapable of reading anything useful. For example one experiment in AC erasure involved driving the write head with a 40 MHz square wave with an initial current of 12 mA which was dropped in 2 mA steps to a final level of 2 mA in successive passes, an order of magnitude more than the usual write current which ranges from high microamps to low milliamps. Any remnant bit patterns left by this erasing process were far too faint to be detected by the read head, but could still be observed using MFM [8].

Even with a DC erasure process, traces of the previously recorded signal may persist until the applied DC field is several times the media coercivity [9].

Deviations in the position of the drive head from the original track may leave significant portions of the previous data along the track edge relatively untouched. Newly written data, present as wide alternating light and dark bands in MFM and STM images, are often superimposed over previously recorded data which persists at the track edges. Regions where the old and new data coincide create continuous magnetization between the two. However, if the new transition is out of phase with the previous one, a few microns of erase band with no definite magnetization are created at the juncture of the old and new tracks. The write field in the erase band is above the coercivity of the media and would change the magnetization in these areas, but its magnitude is not high enough to create new well- defined transitions. One experiment involved writing a fixed pattern of all 1's with a bit interval of 2.5 µm, moving the write head off-track by approximately half a track width, and then writing the pattern again with a frequency slightly higher than that of the previously recorded track for a bit interval of 2.45 µm to create all possible phase differences between the transitions in the old and new tracks. Using a 4.2 µm wide head produced an erase band of approximately 1 µm in width when the old and new tracks were 180° out of phase, dropping to almost nothing when the two tracks were in-phase. Writing data at a higher frequency with the original tracks bit interval at 0.5 µm and the new tracks bit interval at 0.49 µm allows a single MFM image to contain all possible phase differences, showing a dramatic increase in the width of the erase band as the two tracks move from in-phase to 180° out of phase [10].

In addition, the new track width can exhibit modulation which depends on the phase relationship between the old and new patterns, allowing the previous data to be recovered even if the old data patterns themselves are no longer distinct. The overwrite performance also depends on the position of the write head relative to the originally written track. If the head is directly aligned with the track, overwrite performance is relatively good; as the head moves offtrack, the performance drops markedly as the remnant components of the original data are

read back along with the newly-written signal. This effect is less noticeable as the write frequency increases due to the greater attenuation of the field with distance [11].

When all the above factors are combined it turns out that each track contains an image of everything ever written to it, but that the contribution from each "layer" gets progressively smaller the further back it was made. Intelligence organisations have a lot of expertise in recovering these palimpsestuous images.

3. Erasure of Data stored on Magnetic Media

The general concept behind an overwriting scheme is to flip each magnetic domain on the disk back and forth as much as possible (this is the basic idea behind degaussing) without writing the same pattern twice in a row. If the data was encoded directly, we could simply choose the desired overwrite pattern of ones and zeroes and write it repeatedly. However, disks generally use some form of run-length limited (RLL) encoding, so that the adjacent ones won't be written. This encoding is used to ensure that transitions aren't placed too closely together, or too far apart, which would mean the drive would lose track of where it was in the data.

To erase magnetic media, we need to overwrite it many times with alternating patterns in order to expose it to a magnetic field oscillating fast enough that it does the desired flipping of the magnetic domains in a reasonable amount of time. Unfortunately, there is a complication in that we need to saturate the disk surface to the greatest depth possible, and very high frequency signals only "scratch the surface" of the magnetic medium (this phenomenon was used to good effect when HiFi VCRs were introduced by writing the stereo FM audio signal at a lower frequency beneath the higher-frequency video signal, a technique known as depth multiplex recording). Disk drive manufacturers, in trying to achieve ever-higher densities, use the highest possible frequencies, whereas we really require the lowest frequency a disk drive can produce. Even this is still rather high. The best we can do is to use the lowest frequency possible for overwrites, to penetrate as deeply as possible into the recording medium.

The write frequency also determines how effectively previous data can be overwritten due to the dependence of the field needed to cause magnetic switching on the length of time the field is applied. Tests on a number of typical disk drive heads have shown a difference of up to 20 dB in overwrite performance when data recorded at 40 kFCI (flux changes per inch), typical of recent disk drives, is overwritten with a signal varying from 0 to 100 kFCI. The best average performance for the various heads appears to be with an overwrite signal of around 10 kFCI, with the worst performance being at 100 kFCI [12]. The track write width is also affected by the write frequency - as the frequency increases, the write width decreases for both MR and TFI heads. In [13] there was a decrease in write width of around 20% as the write frequency was increased from

1 to 40 kFCI, with the decrease being most marked at the high end of the frequency range. However, the decrease in write width is balanced by a corresponding increase in the two side- erase bands so that the sum of the two remains nearly constant with frequency and equal to the DC erase width for the head. The media coercivity also affects the width of the write and erase bands, with their width dropping as the coercivity increases (this is one of the explanations for the ever-increasing coercivity of newer, higher-density drives).

To try to write the lowest possible frequency we must determine what decoded data to write to produce a low-frequency encoded signal.

In order to understand the theory behind the choice of data patterns to write, it is necessary to take a brief look at the recording methods used in disk drives. The main limit on recording density is that as the bit density is increased, the peaks in the analog signal recorded on the media are read at a rate which may cause them to appear to overlap, creating intersymbol interference which leads to data errors. Traditional peak detector read channels try to reduce the possibility of intersymbol interference by coding data in such a way that the analog signal peaks are separated as far as possible. The read circuitry can then accurately detect the peaks (actually the head itself only detects transitions in magnetisation, so the simplest recording code uses a transition to encode a 1 and the absence of a transition to encode a 0. The transition causes a positive/negative peak in the head output voltage (thus the name "peak detector read channel"). To recover the data, we differentiate the output and look for the zero crossings). Since a long string of 0's will make clocking difficult, we need to set a limit on the maximum consecutive number of 0's. The separation of peaks is implemented as some form of run-length-limited, or RLL, coding.

The RLL encoding used in most current drives is described by pairs of run-length limits (*d, k*), where *d* is the minimum number of 0 symbols which must occur between each 1 symbol in the encoded data, and *k* is the maximum. The parameters (*d, k*) are chosen to place adjacent 1's far enough apart to avoid problems with intersymbol interference, but not so far apart that we lose synchronisation.

The grandfather of all RLL codes was FM, which wrote one user data bit followed by one clock bit, so that a 1 bit was encoded as two transitions (1 wavelength) while a 0 bit was encoded as one transition (« wavelength). A different approach was taken in modified FM (MFM), which suppresses the clock bit except between adjacent 0's (the ambiguity in the use of the term MFM is unfortunate. From here on it will be used to refer to modified FM rather than magnetic force microscopy). Taking three example sequences 0000, 1111, and 1010, these will be encoded as 0(1)0(1)0(1)0, 1(0)1(0)1(0)1, and 1(0)0(0)1(0)0 (where the ()s are the clock bits inserted by the encoding process). The maximum time between 1 bits is now three 0 bits (so that the peaks are no more than four encoded time periods apart), and there is always at least one 0 bit (so that the peaks in the analog signal are at

least two encoded time periods apart), resulting in a (1,3) RLL code. (1,3) RLL/MFM is the oldest code still in general use today, but is only really used in floppy drives which need to remain backwards-compatible.

These constraints help avoid intersymbol interference, but the need to separate the peaks reduces the recording density and therefore the amount of data which can be stored on a disk. To increase the recording density, MFM was gradually replaced by (2,7) RLL (the original "RLL" format), and that in turn by (1,7) RLL, each of which placed less constraints on the recorded signal.

Using our knowledge of how the data is encoded, we can now choose which decoded data patterns to write in order to obtain the desired encoded signal. The three encoding methods described above cover the vast majority of magnetic disk drives. However, each of these has several possible variants. With MFM, only one is used with any frequency, but the newest (1,7) RLL code has at least half a dozen variants in use. For MFM with at most four bit times between transitions, the lowest write frequency possible is attained by writing the repeating decoded data patterns 1010 and 0101. These have a 1 bit every other "data" bit, and the intervening "clock" bits are all 0. We would also like patterns with every other clock bit set to 1 and all others set to 0, but these are not possible in the MFM encoding (such "violations" are used to generate special marks on the disk to identify sector boundaries). The best we can do here is three bit times between transitions, which is generated by repeating the decoded patterns 100100, 010010 and 001001. We should use several passes with these patterns, as MFM drives are the oldest, lowest-density drives around (this is especially true for the very-low-density floppy drives). As such, they are the easiest to recover data from with modern equipment and we need to take the most care with them.

From MFM we jump to the next simplest case, which is (1,7) RLL. Although there can be as many as 8 bit times between transitions, the lowest sustained frequency we can have in practice is 6 bit times between transitions. This is a desirable property from the point of view of the clock-recovery circuitry, and all (1,7) RLL codes seem to have this property. We now need to find a way to write the desired pattern without knowing the particular (1,7) RLL code used. We can do this by looking at the way the drives error-correction system works. The error- correction is applied to the decoded data, even though errors generally occur in the encoded data. In order to make this work well, the data encoding should have limited error amplification, so that an erroneous encoded bit should affect only a small, finite number of decoded bits.

Decoded bits therefore depend only on nearby encoded bits, so that a repeating pattern of encoded bits will correspond to a repeating pattern of decoded bits. The repeating pattern of encoded bits is 6 bits long. Since the rate of the code is 2/3, this corresponds to a repeating pattern of 4 decoded bits. There are only 16 possibilities for this pattern, making it feasible to write all of them during the erase process. So to achieve good overwriting of (1,7) RLL disks, we write the patterns

0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, and 1111. These patterns also conveniently cover two of the ones needed for MFM overwrites, although we should add a few more iterations of the MFM-specific patterns for the reasons given above.

Finally, we have (2,7) RLL drives. These are similar to MFM in that an eight-bit-time signal can be written in some phases, but not all. A six-bit-time signal will fill in the remaining cracks. Using a « encoding rate, an eight-bit-time signal corresponds to a repeating pattern of 4 data bits. The most common (2,7) RLL code is shown below:

The most common (2,7) RLL Code	
Decoded Data	**(2,7) RLL Encoded Data**
00	1000
01	0100
100	001000
101	100100
111	000100
1100	00001000
1101	00100100

The second most common (2,7) RLL code is the same but with the "decoded data" complemented, which doesn't alter these patterns. Writing the required encoded data can be achieved for every other phase using patterns of 0x33, 0x66, 0xCC and 0x99, which are already written for (1,7) RLL drives.

Six-bit-time patterns can be written using 3-bit repeating patterns. The all-zero and all-one patterns overlap with the (1,7) RLL patterns, leaving six others:

```
001001001001001001001001
   2   4   9   2   4   9
```

in binary or 0x24 0x92 0x49, 0x92 0x49 0x24 and 0x49 0x24 0x92 in hex, and

```
011011011011011011011011
   6   D   B   6   D   B
```

in binary or 0x6D 0xB6 0xDB, 0xB6 0xDB 0x6D and 0xDB 0x6D 0xB6 in hex. The first three are the same as the MFM patterns, so we need only three extra patterns to cover (2,7) RLL drives.

Although (1,7) is more popular in recent (post-1990) drives, some older hard drives do still use (2,7) RLL, and with the ever-increasing reliability of newer

drives it is likely that they will remain in use for some time to come, often being passed down from one machine to another. The above three patterns also cover any problems with endianness issues, which weren't a concern in the previous two cases, but would be in this case (actually, thanks to the strong influence of IBM mainframe drives, everything seems to be uniformly big-endian within bytes, with the most significant bit being written to the disk first).

The latest high-density drives use methods like Partial-Response Maximum-Likelihood (PRML) encoding, which may be roughly equated to the trellis encoding done by V.32 modems in that it is effective but computationally expensive. PRML codes are still RLL codes, but with somewhat different constraints. A typical code might have (0,4,4) constraints in which the 0 means that 1's in a data stream can occur right next to 0's (so that peaks in the analog readback signal are not separated), the first 4 means that there can be no more than four 0's between 1's in a data stream, and the second 4 specifies the maximum number of 0's between 1's in certain symbol subsequences. PRML codes avoid intersymbol influence errors by using digital filtering techniques to shape the read signal to exhibit desired frequency and timing characteristics (this is the "partial response" part of PRML) followed by maximum- likelihood digital data detection to determine the most likely sequence of data bits that was written to the disk (this is the "maximum likelihood" part of PRML). PRML channels achieve the same low bit error rate as standard peak-detection methods, but with much higher recording densities, while using the same heads and media. Several manufacturers are currently engaged in moving their peak-detection-based product lines across to PRML, giving a 30-40% density increase over standard RLL channels [14].

Since PRML codes don't try to separate peaks in the same way that non-PRML RLL codes do, all we can do is to write a variety of random patterns because the processing inside the drive is too complex to second- guess. Fortunately, these drives push the limits of the magnetic media much more than older drives ever did by encoding data with much smaller magnetic domains, closer to the physical capacity of the magnetic media (the current state of the art in PRML drives has a track density of around 6700 TPI (tracks per inch) and a data recording density of 170 kFCI, nearly double that of the nearest (1,7) RLL equivalent. A convenient side-effect of these very high recording densities is that a written transition may experience the write field cycles for successive transitions, especially at the track edges where the field distribution is much broader [15]. Since this is also where remnant data is most likely to be found, this can only help in reducing the recoverability of the data). If these drives require sophisticated signal processing just to read the most recently written data, reading overwritten layers is also correspondingly more difficult. A good scrubbing with random data will do about as well as can be expected.

We now have a set of 22 overwrite patterns which should erase everything, regardless of the raw encoding. The basic disk eraser can be improved slightly by adding random passes before and after the erase process, and by performing the

deterministic passes in random order to make it more difficult to guess which of the known data passes were made at which point. To deal with all this in the overwrite process, we use the sequence of 35 consecutive writes shown below:

Overwrite Data				
Pass No.	**Data Written**	**Encoding Scheme Targeted**		
1	Random			
2	Random			
3	Random			
4	Random			
5	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
6	10101010 10101010 10101010 0xAA	(1,7) RLL		MFM
7	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
8	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
9	00100100 10010010 01001001 0x24 0x92 0x49		(2,7) RLL	MFM
10	00000000 00000000 00000000 0x00	(1,7) RLL	(2,7) RLL	
11	00010001 00010001 00010001 0x11	(1,7) RLL		
12	00100010 00100010 00100010 0x22	(1,7) RLL		
13	00110011 00110011 00110011 0x33	(1,7) RLL	(2,7) RLL	
14	01000100 01000100 01000100 0x44	(1,7) RLL		
15	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
16	01100110 01100110 01100110 0x66	(1,7) RLL	(2,7) RLL	
17	01110111 01110111 01110111 0x77	(1,7) RLL		
18	10001000 10001000 10001000 0x88	(1,7) RLL		
19	10011001 10011001 10011001 0x99	(1,7) RLL	(2,7) RLL	
20	10101010 10101010 10101010 0xAA	(1,7) RLL		MFM
21	10111011 10111011 10111011 0xBB	(1,7) RLL		
22	11001100 11001100 11001100 0xCC	(1,7) RLL	(2,7) RLL	
23	11011101 11011101 11011101 0xDD	(1,7) RLL		

24	11101110 11101110 11101110 0xEE	(1,7) RLL		
25	11111111 11111111 11111111 0xFF	(1,7) RLL	(2,7) RLL	
26	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
27	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
28	00100100 10010010 01001001 0x24 0x92 0x49		(2,7) RLL	MFM
29	01101101 10110110 11011011 0x6D 0xB6 0xDB		(2,7) RLL	
30	10110110 11011011 01101101 0xB6 0xDB 0x6D		(2,7) RLL	
31	11011011 01101101 10110110 0xDB 0x6D 0xB6		(2,7) RLL	
32	Random			
33	Random			
34	Random			
35	Random			

The MFM-specific patterns are repeated twice because MFM drives have the lowest density and are thus particularly easy to examine. The deterministic patterns between the random writes are permuted before the write is performed, to make it more difficult for an opponent to use knowledge of the erasure data written to attempt to recover overwritten data (in fact we need to use a cryptographically strong random number generator to perform the permutations to avoid the problem of an opponent who can read the last overwrite pass being able to predict the previous passes and "echo cancel" passes by subtracting the known overwrite data).

If the device being written to supports caching or buffering of data, this should be disabled to ensure that physical disk writes are performed for each pass instead of everything but the last pass being lost in the buffering. For example physical disk access can be forced during SCSI-2 Group 1 write commands by setting the Force Unit Access bit in the SCSI command block (although at least one popular drive has a bug which causes all writes to be ignored when this bit is set - remember to test your overwrite scheme before you deploy it). Another consideration which needs to be taken into account when trying to erase data through software is that drives conforming to some of the higher-level protocols such as the various SCSI standards are relatively free to interpret commands sent to them in whichever way they choose (as long as they still conform to the SCSI specification). Thus some

drives, if sent a FORMAT UNIT command may return immediately without performing any action, may simply perform a read test on the entire disk (the most common option), or may actually write data to the disk (the SCSI- 2 standard includes an initialization pattern (IP) option for the FORMAT UNIT command, however this is not necessarily supported by existing drives).

If the data is very sensitive and is stored on floppy disk, it can best be destroyed by removing the media from the disk liner and burning it, or by burning the entire disk, liner and all (most floppy disks burn remarkably well - albeit with quantities of oily smoke - and leave very little residue).

4. Other Methods of Erasing Magnetic Media

The previous section has concentrated on erasure methods which require no specialised equipment to perform the erasure. Alternative means of erasing media which do require specialised equipment are degaussing (a process in which the recording media is returned to its initial state) and physical destruction. Degaussing is a reasonably effective means of purging data from magnetic disk media, and will even work through most drive cases (research has shown that the aluminium housings of most disk drives attenuate the degaussing field by only about 2 dB [16]).

The switching of a single-domain magnetic particle from one magnetization direction to another requires the overcoming of an energy barrier, with an external magnetic field helping to lower this barrier. The switching depends not only on the magnitude of the external field, but also on the length of time for which it is applied. For typical disk drive media, the short-term field needed to flip enough of the magnetic domains to be useful in recording a signal is about 1/3 higher than the coercivity of the media (the exact figure varies with different media types) [17].

However, to effectively erase a medium to the extent that recovery of data from it becomes uneconomical requires a magnetic force of about five times the coercivity of the medium [18], although even small external magnetic fields are sufficient to upset the normal operation of a hard disk (typically a few gauss at DC, dropping to a few milligauss at 1 MHz). Coercivity (measured in Oersteds, Oe) is a property of magnetic material and is defined as the amount of magnetic field necessary to reduce the magnetic induction in the material to zero - the higher the coercivity, the harder it is to erase data from a medium. Typical figures for various types of magnetic media are given below:

Typical Media Coercivity Figures	
Medium	**Coercivity**
5.25" 360K floppy disk	300 Oe

5.25" 1.2M floppy disk	675 Oe
3.5" 720K floppy disk	300 Oe
3.5" 1.44M floppy disk	700 Oe
3.5" 2.88M floppy disk	750 Oe
3.5" 21M floptical disk	750 Oe
Older (1980's) hard disks	900-1400 Oe
Newer (1990's) hard disks	1400-2200 Oe
1/2" magnetic tape	300 Oe
1/4" QIC tape	550 Oe
8 mm metallic particle tape	1500 Oe
DAT metallic particle tape	1500 Oe

US Government guidelines class tapes of 350 Oe coercivity or less as low-energy or Class I tapes and tapes of 350-750 Oe coercivity as high-energy or Class II tapes. Degaussers are available for both types of tapes. Tapes of over 750 Oe coercivity are referred to as Class III, with no known degaussers capable of fully erasing them being known [19], since even the most powerful commercial AC degausser cannot generate the recommended 7,500 Oe needed for full erasure of a typical DAT tape currently used for data backups.

Degaussing of disk media is somewhat more difficult - even older hard disks generally have a coercivity equivalent to Class III tapes, making them fairly difficult to erase at the outset. Since manufacturers rate their degaussers in peak gauss and measure the field at a certain orientation which may not be correct for the type of medium being erased, and since degaussers tend to be rated by whether they erase sufficiently for clean rerecording rather than whether they make the information impossible to recover, it may be necessary to resort to physical destruction of the media to completely sanitise it (in fact since degaussing destroys the sync bytes, ID fields, error correction information, and other paraphernalia needed to identify sectors on the media, thus rendering the drive unusable, it makes the degaussing process mostly equivalent to physical destruction). In addition, like physical destruction, it requires highly specialised equipment which is expensive and difficult to obtain (one example of an adequate degausser was the 2.5 MW Navy research magnet used by a former Pentagon site manager to degauss a 14" hard drive for 1« minutes. It bent the platters on the drive and probably succeeded in erasing it beyond the capabilities of any data recovery attempts [20]).

5. Further Problems with Magnetic Media

A major issue which cannot be easily addressed using any standard software-based overwrite technique is the problem of defective sector handling. When the drive is manufactured, the surface is scanned for defects which are added to a defect list or flaw map. If further defects, called grown defects, occur during the life of the drive, they are added to the defect list by the drive or by drive management software. There are several techniques which are used to mask the defects in the defect list. The first, alternate tracks, moves data from tracks with defects to known good tracks. This scheme is the simplest, but carries a high access cost, as each read from a track with defects requires seeking to the alternate track and a rotational latency delay while waiting for the data location to appear under the head, performing the read or write, and, if the transfer is to continue onto a neighbouring track, seeking back to the original position. Alternate tracks may be interspersed among data tracks to minimise the seek time to access them.

A second technique, alternate sectors, allocates alternate sectors at the end of the track to minimise seeks caused by defective sectors. This eliminates the seek delay, but still carries some overhead due to rotational latency. In addition it reduces the usable storage capacity by 1-3%.

A third technique, inline sector sparing, again allocates a spare sector at the end of each track, but resequences the sector ID's to skip the defective sector and include the spare sector at the end of the track, in effect pushing the sectors past the defective one towards the end of the track. The associated cost is the lowest of the three, being one sector time to skip the defective sector [21].

The handling of mapped-out sectors and tracks is an issue which can't be easily resolved without the cooperation of hard drive manufacturers. Although some SCSI and IDE hard drives may allow access to defect lists and even to mapped-out areas, this must be done in a highly manufacturer- and drive-specific manner. For example the SCSI-2 READ DEFECT DATA command can be used to obtain a list of all defective areas on the drive. Since SCSI logical block numbers may be mapped to arbitrary locations on the disk, the defect list is recorded in terms of heads, tracks, and sectors. As all SCSI device addressing is performed in terms of logical block numbers, mapped-out sectors or tracks cannot be addressed. The only reasonably portable possibility is to clear various automatic correction flags in the read-write error recovery mode page to force the SCSI device to report read/write errors to the user instead of transparently remapping the defective areas. The user can then use the READ LONG and WRITE LONG commands (which allow access to sectors and extra data even in the presence of read/write errors), to perform any necessary operations on the defective areas, and then use the REASSIGN BLOCKS command to reassign the defective sections. However this operation requires an in-depth knowledge of the operation of the SCSI device and extensive changes to disk drivers, and more or less defeats the purpose of having an intelligent peripheral.

The ANSI X3T-10 and X3T-13 subcommittees are currently looking at creating new standards for a Universal Security Reformat command for IDE and SCSI hard disks which will address these issues. This will involve a multiple-pass overwrite process which covers mapped-out disk areas with deliberate off-track writing. Many drives available today can be modified for secure erasure through a firmware upgrade, and once the new firmware is in place the erase procedure is handled by the drive itself, making unnecessary any interaction with the host system beyond the sending of the command which begins the erase process.

Long-term ageing can also have a marked effect on the erasability of magnetic media. For example, some types of magnetic tape become increasingly difficult to erase after being stored at an elevated temperature or having contained the same magnetization pattern for a considerable period of time [22]. The same applies for magnetic disk media, with decreases in erasability of several dB being recorded [23]. The erasability of the data depends on the amount of time it has been stored on the media, not on the age of the media itself (so that, for example, a five-year-old freshly-written disk is no less erasable than a new freshly-written disk).

The dependence of media coercivity on temperature can affect overwrite capability if the data was initially recorded at a temperature where the coercivity was low (so that the recorded pattern penetrated deep into the media), but must be overwritten at a temperature where the coercivity is relatively high. This is important in hard disk drives, where the temperature varies depending on how long the unit has been used and, in the case of drives with power-saving features enabled, how recently and frequently it has been used. However the overwrite performance depends not only on temperature-dependent changes in the media, but also on temperature-dependent changes in the read/write head. Thankfully the combination of the most common media used in current drives with various common types of read/write heads produce a change in overwrite performance of only a few hundredths of a decibel per degree over the temperature range -40°C to + 40°C, as changes in the head compensate for changes in the media [24].

Another issue which needs to be taken into account is the ability of most newer storage devices to recover from having a remarkable amount of damage inflicted on them through the use of various error-correction schemes. As increasing storage densities began to lead to multiple-bit errors, manufacturers started using sophisticated error-correction codes (ECC's) capable of correcting multiple error bursts. A typical drive might have 512 bytes of data, 4 bytes of CRC, and 11 bytes of ECC per sector. This ECC would be capable of correcting single burst errors of up to 22 bits or double burst errors of up to 11 bits, and can detect a single burst error of up to 51 bits or three burst errors of up to 11 bits in length [25]. Another drive manufacturer quotes the ability to correct up to 120 bits, or up to 32 bits on the fly, using 198-bit Reed-Solomon ECC [26]. Therefore even if some data is reliably erased, it may be possible to recover it using the built-in error-correction capabilities of the drive. Conversely, any erasure scheme which manages to destroy the ECC information (for example through the use of the SCSI-2 WRITE

LONG command which can be used to write to areas of a disk sector outside the normal data areas) stands a greater chance of making the data unrecoverable.

6. Sidestepping the Problem

The easiest way to solve the problem of erasing sensitive information from magnetic media is to ensure that it never gets to the media in the first place. Although not practical for general data, it is often worthwhile to take steps to keep particularly important information such as encryption keys from ever being written to disk. This would typically happen when the memory containing the keys is paged out to disk by the operating system, where they can then be recovered at a later date, either manually or using software which is aware of the in-memory data format and can locate it automatically in the swap file (for example there exists software which will search the Windows swap file for keys from certain DOS encryption programs). An even worse situation occurs when the data is paged over a network, allowing anyone with a packet sniffer or similar tool on the same subnet to observe the information (for example there exists software which will monitor and even alter NFS traffic on the fly which could be modified to look for known in-memory data patterns moving to and from a networked swap disk [27]).

To solve these problems the memory pages containing the information can be locked to prevent them from being paged to disk or transmitted over a network. This approach is taken by at least one encryption library, which allocates all keying information inside protected memory blocks visible to the user only as opaque handles, and then optionally locks the memory (provided the underlying OS allows it) to prevent it from being paged [28]. The exact details of locking pages in memory depend on the operating system being used. Many Unix systems now support the `mlock()`/`munlock()` calls or have some alternative mechanism hidden among the `mmap()`-related functions which can be used to lock pages in memory. Unfortunately these operations require superuser privileges because of their potential impact on system performance if large ranges of memory are locked. Other systems such as Microsoft Windows NT allow user processes to lock memory with the `VirtualLock()`/`VirtualUnlock()` calls, but limit the total number of regions which can be locked.

Most paging algorithms are relatively insensitive to having sections of memory locked, and can even relocate the locked pages (since the logical to physical mapping is invisible to the user), or can move the pages to a "safe" location when the memory is first locked. The main effect of locking pages in memory is to increase the minimum working set size which, taken in moderation, has little noticeable effect on performance. The overall effects depend on the operating system and/or hardware implementations of virtual memory. Most Unix systems have a global page replacement policy in which a page fault may be satisfied by any page frame. A smaller number of operating systems use a local page replacement policy in which pages are allocated from a fixed (or occasionally dynamically variable) number of page frames allocated on a per- process basis.

This makes them much more sensitive to the effects of locking pages, since every locked page decreases the (finite) number of pages available to the process. On the other hand it makes the system as a whole less sensitive to the effects of one process locking a large number of pages. The main effective difference between the two is that under a local replacement policy a process can only lock a small fixed number of pages without affecting other processes, whereas under a global replacement policy the number of pages a process can lock is determined on a system-wide basis and may be affected by other processes.

In practice neither of these allocation strategies seem to cause any real problems. Although any practical measurements are very difficult to perform since they vary wildly depending on the amount of physical memory present, paging strategy, operating system, and system load, in practice locking a dozen 1K regions of memory (which might be typical of a system on which a number of users are running programs such as mail encryption software) produced no noticeable performance degradation observable by system- monitoring tools. On machines such as network servers handling large numbers of secure connections (for example an HTTP server using SSL), the effects of locking large numbers of pages may be more noticeable.

7. Methods of Recovery for Data stored in Random-Access Memory

Contrary to conventional wisdom, "volatile" semiconductor memory does not entirely lose its contents when power is removed. Both static (SRAM) and dynamic (DRAM) memory retains some information on the data stored in it while power was still applied. SRAM is particularly susceptible to this problem, as storing the same data in it over a long period of time has the effect of altering the preferred power-up state to the state which was stored when power was removed. Older SRAM chips could often "remember" the previously held state for several days. In fact, it is possible to manufacture SRAM's which always have a certain state on power-up, but which can be overwritten later on - a kind of "writeable ROM".

DRAM can also "remember" the last stored state, but in a slightly different way. It isn't so much that the charge (in the sense of a voltage appearing across a capacitance) is retained by the RAM cells, but that the thin oxide which forms the storage capacitor dielectric is highly stressed by the applied field, or is not stressed by the field, so that the properties of the oxide change slightly depending on the state of the data. One thing that can cause a threshold shift in the RAM cells is ionic contamination of the cell(s) of interest, although such contamination is rarer now than it used to be because of robotic handling of the materials and because the purity of the chemicals used is greatly improved. However, even a perfect oxide is subject to having its properties changed by an applied field. When it comes to contaminants, sodium is the most common offender - it is found virtually everywhere, and is a fairly small (and therefore mobile) atom with a

positive charge. In the presence of an electric field, it migrates towards the negative pole with a velocity which depends on temperature, the concentration of the sodium, the oxide quality, and the other impurities in the oxide such as dopants from the processing. If the electric field is zero and given enough time, this stress tends to dissipate eventually.

The stress on the cell is a cumulative effect, much like charging an RC circuit. If the data is applied for only a few milliseconds then there is very little "learning" of the cell, but if it is applied for hours then the cell will acquire a strong (relatively speaking) change in its threshold. The effects of the stress on the RAM cells can be measured using the built-in self test capabilities of the cells, which provide the ability to impress a weak voltage on a storage cell in order to measure its margin. Cells will show different margins depending on how much oxide stress has been present. Many DRAM's have undocumented test modes which allow some normal I/O pin to become the power supply for the RAM core when the special mode is active. These test modes are typically activated by running the RAM in a nonstandard configuration, so that a certain set of states which would not occur in a normally-functioning system has to be traversed to activate the mode. Manufacturers won't admit to such capabilities in their products because they don't want their customers using them and potentially rejecting devices which comply with their spec sheets, but have little margin beyond that.

A simple but somewhat destructive method to speed up the annihilation of stored bits in semiconductor memory is to heat it. Both DRAM's and SRAM's will lose their contents a lot more quickly at Tjunction = 140°C than they will at room temperature. Several hours at this temperature with no power applied will clear their contents sufficiently to make recovery difficult. Conversely, to extend the life of stored bits with the power removed, the temperature should be dropped below -60°C. Such cooling should lead to weeks, instead of hours or days, of data retention.

8. Erasure of Data stored in Random-Access Memory

Simply repeatedly overwriting the data held in DRAM with new data isn't nearly as effective as it is for magnetic media. The new data will begin stressing or relaxing the oxide as soon as it is written, and the oxide will immediately begin to take a "set" which will either reinforce the previous "set" or will weaken it. The greater the amount of time that new data has existed in the cell, the more the old stress is "diluted", and the less reliable the information extraction will be. Generally, the rates of change due to stress and relaxation are in the same order of magnitude. Thus, a few microseconds of storing the opposite data to the currently stored value will have little effect on the oxide. Ideally, the oxide should be exposed to as much stress at the highest feasible temperature and for as long as possible to get the greatest "erasure" of the data. Unfortunately if carried too far this has a rather

detrimental effect on the life expectancy of the RAM.

Therefore the goal to aim for when sanitising memory is to store the data for as long as possible rather than trying to change it as often as possible. Conversely, storing the data for as short a time as possible will reduce the chances of it being "remembered" by the cell. Based on tests on DRAM cells, a storage time of one second causes such a small change in threshold that it probably isn't detectable. On the other hand, one minute is probably detectable, and 10 minutes is certainly detectable.

The most practical solution to the problem of DRAM data retention is therefore to constantly flip the bits in memory to ensure that a memory cell never holds a charge long enough for it to be "remembered". While not practical for general use, it is possible to do this for small amounts of very sensitive data such as encryption keys. This is particularly advisable where keys are stored in the same memory location for long periods of time and control access to large amounts of information, such as keys used for transparent encryption of files on disk drives. The bit-flipping also has the convenient side-effect of keeping the page containing the encryption keys at the top of the queue maintained by the system's paging mechanism, greatly reducing the chances of it being paged to disk at some point.

9. Conclusion

Data overwritten once or twice may be recovered by subtracting what is expected to be read from a storage location from what is actually read. Data which is overwritten an arbitrarily large number of times can still be recovered provided that the new data isn't written to the same location as the original data (for magnetic media), or that the recovery attempt is carried out fairly soon after the new data was written (for RAM). For this reason it is effectively impossible to sanitise storage locations by simple overwriting them, no matter how many overwrite passes are made or what data patterns are written. However by using the relatively simple methods presented in this paper the task of an attacker can be made significantly more difficult, if not prohibitively expensive.

Epilogue

In the time since this paper was published, some people have treated the 35-pass overwrite technique described in it more as a kind of voodoo incantation to banish evil spirits than the result of a technical analysis of drive encoding techniques. As a result, they advocate applying the voodoo to PRML and EPRML drives even though it will have no more effect than a simple scrubbing with random data. In fact performing the full 35-pass overwrite is pointless for any drive since it targets a blend of scenarios involving all types of (normally-used) encoding technology, which covers everything back to 30+-year-old MFM methods (if you don't understand that statement, re-read the paper). If you're using a drive which uses encoding technology X, you only need to perform the passes specific to X, and you

never need to perform all 35 passes. For any modern PRML/EPRML drive, a few passes of random scrubbing is the best you can do. As the paper says, "A good scrubbing with random data will do about as well as can be expected". This was true in 1996, and is still true now.

Acknowledgments

The author would like to thank Nigel Bree, Peter Fenwick, Andy Hospodor, Kevin Martinez, Colin Plumb, and Charles Preston for their advice and input during the preparation of this paper.

References

[1] "Emergency Destruction of Information Storing Media", M.Slusarczuk et al, Institute for Defense Analyses, December 1987.

[2] "A Guide to Understanding Data Remanence in Automated Information Systems", National Computer Security Centre, September 1991.

[3] "Detection of Digital Information from Erased Magnetic Disks", Venugopal Veeravalli, Masters thesis, Carnegie-Mellon University, 1987.

[4] "Magnetic force microscopy: General principles and application to longitudinal recording media", D.Rugar, H.Mamin, P.Guenther, S.Lambert, J.Stern, I.McFadyen, and T.Yogi, *Journal of Applied Physics*, **Vol.68, No.3** (August 1990), p.1169.

[5] "Tunneling-stabilized Magnetic Force Microscopy of Bit Tracks on a Hard Disk", Paul Rice and John Moreland, *IEEE Trans.on Magnetics*, **Vol.27, No.3** (May 1991), p.3452.

[6] "NanoTools: The Homebrew STM Page", Jim Rice, NanoTools: The Homebrew STM Page (now defunct). This page became Angstrom Tools LLC, the best equivalent of the old NanoTools page is the General STM Info page.

[7] "Magnetic Force Scanning Tunnelling Microscope Imaging of Overwritten Data", Romel Gomez, Amr Adly, Isaak Mayergoyz, Edward Burke, *IEEE Trans.on Magnetics*, **Vol.28, No.5** (September 1992), p.3141.

[8] "Comparison of Magnetic Fields of Thin-Film Heads and Their Corresponding Patterns Using Magnetic Force Microscopy", Paul Rice, Bill Hallett, and John Moreland, *IEEE Trans.on Magnetics*, **Vol.30, No.6** (November 1994), p.4248.

[9] "Computation of Magnetic Fields in Hysteretic Media", Amr Adly, Isaak Mayergoyz, Edward Burke, *IEEE Trans.on Magnetics*, **Vol.29, No.6** (November 1993), p.2380.

[10] "Magnetic Force Microscopy Study of Edge Overwrite Characteristics in Thin Film Media", Jian- Gang Zhu, Yansheng Luo, and Juren Ding, *IEEE Trans.on Magnetics*, **Vol.30, No.6** (November 1994), p.4242.

[11] "Microscopic Investigations of Overwritten Data", Romel Gomez, Edward Burke, Amr Adly, Isaak Mayergoyz, J.Gorczyca, *Journal of Applied Physics*, **Vol.73, No.10** (May 1993), p.6001.

[12] "Relationship between Overwrite and Transition Shift in Perpendicular Magnetic Recording", Hiroaki Muraoka, Satoshi Ohki, and Yoshihisa Nakamura, *IEEE Trans.on Magnetics*, **Vol.30, No.6** (November 1994), p.4272.

[13] "Effects of Current and Frequency on Write, Read, and Erase Widths for Thin-Film Inductive and Magnetoresistive Heads", Tsann Lin, Jodie Christner, Terry Mitchell, Jing-Sheng Gau, and Peter George, *IEEE Trans.on Magnetics*, **Vol.25, No.1** (January 1989), p.710.

[14] "PRML Read Channels: Bringing Higher Densities and Performance to New-Generation Hard Drives", Quantum Corporation, 1995.

[15] "Density and Phase Dependence of Edge Erase Band in MR/Thin Film Head Recording", Yansheng Luo, Terence Lam, Jian-Gang Zhu, *IEEE Trans.on Magnetics*, **Vol.31, No.6** (November 1995), p.3105.

[16] "A Guide to Understanding Data Remanence in Automated Information Systems", National Computer Security Centre, September 1991.

[17] "Time-dependant Magnetic Phenomena and Particle-size Effects in Recording Media", *IEEE Trans.on Magnetics*, **Vol.26, No.1** (January 1990), p.193.

[18] "The Data Dilemna", Charles Preston, Security Management Journal, February 1995.

[19] "Magnetic Tape Degausser", NSA/CSS Specification L14-4-A, 31 October 1985.

[20] "How many times erased does DoD want?", David Hayes, posting to comp.periphs.scsi newsgroup, 24 July 1991, message-ID 1991Jul24.050701.16005@sulaco.lone star.org.

[21] "The Changing Nature of Disk Controllers", Andrew Hospodor and Albert Hoagland, *Proceedings of the IEEE*, **Vol.81, No.4** (April 1993), p.586.

[22] "Annealing Study of the Erasability of High Energy Tapes", L.Lekawat, G.Spratt, and M.Kryder, *IEEE Trans.on Magnetics*, **Vol.29, No.6** (November 1993), p.3628.

[23] "The Effect of Aging on Erasure in Particulate Disk Media", K.Mountfield and M.Kryder, *IEEE Trans.on Magnetics*, **Vol.25, No 5** (September 1989), p.3638.

[24] "Overwrite Temperature Dependence for Magnetic Recording", Takayuki Takeda, Katsumichi Tagami, and Takaaki Watanabe, *Journal of Applied Physics*, **Vol.63, No.8** (April 1988), p.3438.

[25] Conner 3.5" hard drive data sheets, 1994, 1995.

[26] "Technology and Time-to-Market: The Two Go Hand-in-Hand", Quantum Corporation, 1995.

[27] "Basic Flaws in Internet Security and Commerce", Paul Gauthier, posting to comp.security.unix newsgroup, 9 October 1995, message-ID gauthier.813274073@espresso.cs.ber keley.edu.

[28] "cryptlib Free Encryption Library", Peter Gutmann, cryptlib.

DS5002FP: Secure Microprocessor Chip

SECTION 1: INTRODUCTION

The Secure Microcontroller family is a line of 8051–compatible devices that utilize nonvolatile RAM (NV RAM) rather than ROM for program storage. The use of NV RAM allows the design of a "soft" microcontroller which provides a number of unique features to embedded system designers. Foremost among these is the enhanced security features that are employed by the Secure Microcontroller Family to protect the user application software against piracy and tampering. These devices offer varying degrees of security, ranging from simple access prevention to a full encryption of program and data memory of the device. Attempts to gain access to protected information will result in the self–destruction of all data. The Secure Microcontroller family is the heart of a wide range of security–critical applications such as electronic banking, commercial transactions, and pay TV access control, or any situation which requires the protection of proprietary software and algorithms.

The Secure Microcontroller family is divided between chips and modules. The chips are monolithic microprocessors that connect to a standard SRAM and lithium battery. The modules combine the microprocessor with the SRAM and lithium battery in a preassembled, pretested module. Depending on the specific configuration, modules are available in either 40–pin encapsulated DIP or SIMM module format.

In addition to NV RAM, Dallas Semiconductor microcontrollers offer a number of peripherals that simplify and reduce the cost of embedded systems. Although the specific features of each chip or module vary, all devices offer the following basic feature set:

- 100% code–compatible with 8051
- Directly addresses 64KB program/64KB data memory
- Nonvolatile memory control circuitry
- 10–year data retention in the absence of power
- In–system reprogramming via serial port
- 128 bytes fast access scratchpad RAM
- Two 16–bit general purpose timer/counters
- One UART
- Five interrupts with two external
- Dedicated memory bus, preserving four 8–bit ports for general purpose I/O
- Power–Fail Reset
- Early Warning Power Fail Interrupt
- Watchdog Timer

SOFTWARE SECURITY

One of the most important features of the Secure Microcontroller family is firmware/memory security. The devices were specifically designed to offer an unprecedented level of protection to the user application software, preventing unauthorized copying of firmware and denying access to critical data values. The use of RAM rather than the traditional ROM or EPROM for program storage increases the security, since tampering with the system will result in the loss of the RAM contents. Additional features such as real–time high–speed memory encryption, generation of dummy addresses on the bus, and internal storage of vector RAM increases the security of a Secure Microcontroller/Microprocessor–based system.

The DS5002FP Secure Microprocessor Chip and DS2252T Secure Microcontroller Module offer the highest level of security, with permanently enabled memory encryption, a 64–bit random encryption key, and a self–destruct input for tamper protection. The DS5000FP Soft Microprocessor Chip and DS5000(T) and DS2250(T) Soft Microcontroller Modules offer lesser, but still substantial, protection with optional data encryption and a 48–bit encryption key.

SEPARATE ADDRESS/DATA BUS

Soft Microprocessor chips provide a non–multiplexed address/data bus that interfaces to memory without interfering with I/O ports. This Byte–wide bus connects directly to standard CMOS SRAM in 8K x 8, 32K x 8, or 128K x 8 densities with no glue logic. Note that this is in addition to the standard 8051 port 0 and 2 multiplexed bus. In module form, the Byte–wide bus is already connected directly to on–board SRAM, so the memory access becomes transparent and the I/O ports free for application use. The extra memory bus also allows for a time–of–day function to be included, and all Soft Microcontroller modules are available with built in real–time clocks. The same clock devices are individually available when building a system from chips. Battery backup and decoding are automatically handled by the microprocessor.

Program Status Flags

All of the Program Status flags are contained in the PSW register. Instructions which affect the states of the flags are summarized below.

INSTRUCTIONS THAT AFFECT FLAG SETTINGS

	FLAGS				FLAGS		
INSTRUCTION	C	OV	AC	INSTRUCTION	C	OV	AC
ADD	↕	↕	↕	CLR C	0		
ADDC	↕	↕	↕	CPL C	↕		
SUBB	↕	↕	↕	ANL C, bit	↕		
MUL	0	↕		ANL C, $\overline{bit}$	↕		
DIV	0	↕		ORL C, bit	↕		
DA	↕			ORL C, $\overline{bit}$	↕		
RRC	↕			MOV C, bit	↕		
RLC	↕			CJNE	↕		
SETB C	1						

LEGEND:
0 = Cleared to 0
1 = Set to a 1
↕ = Modified according to the result of the operation.

SECTION 5: MEMORY INTERCONNECT

The Secure Microcontroller family is divided between chips and modules. This sections illustrates the memory interconnect for the various chips and shows block diagrams of selected modules. The SoftMicroprocessor chips are 80–pin QFP packages that connect to low power CMOS SRAM. The SRAM connection is made through the Byte–wide bus. When using a chip, the user must connect this Byte--wide bus to the RAM as shown in this section. In module form, the bus is connected inside the package. Table 5–1 shows some of the preferred RAM choices. Note that any standard SRAM will work, but data retention lifetime is dependent on RAM data retention current and battery capacity. Lower currents naturally allow the use of smaller batteries. This is covered in detail in Section 6.

RECOMMENDED SRAMs FOR USE WITH SOFT MICROCONTROLLERS Table 5–1

RAM SIZE	VENDOR	PART NUMBER	DATA RETENTION CURRENT 25°C	DATA RETENTION CURRENT 40°C	DATA RETENTION CURRENT 70°C
8K x 8	Dallas	DS2064	0.05 µA	–	–
8K x 8	Sharp	LH5168	–	–	0.6 µA
32K x 8	Hitachi	HM62256LP–SL	–	3 µA	10 µA
32K x 8	Mitsubishi	M5M5256BP–LL	1 µA	–	10 µA
32K x 8	Sony	CXK58257AP–LX	1 µA	2 µA	10 µA
32K x 8	Sony	CXK58527AP–LLX	0.3 µA	0.6 µA	3 µA
128K x 8	Hitachi	HM628128LP–SL	1 µA	–	10 µA
128K x 8	Mitsubishi	M5M51008P–LL	1 µA	–	10 µA
128K x 8	Sony	CXK581000P–LL	1.2 µA	2.4 µA	12 µA

Recommended RAMs are given with the manufacturers specified data retention current at 3V. Missing numbers are conditions unspecified by the manufacturer.

In the case of the DS5000FP, the microprocessor can connect to either one or two SRAMs. They can be 8K bytes or 32K bytes, though the case of two 8K RAMs is unlikely from a cost perspective. Figure 5–1 illustrates the memory connection of a DS5000FP connected to one 32K x 8. $\overline{CE1}$ provides the chip select, and R/$\overline{W}$ supplies the $\overline{WE}$ signal. A second RAM could be added by simply using $\overline{CE2}$ as the chip enable with a common connection for the other signals.

In the case of DS5000 based modules including DS5000(T) and DS2250T, the SRAM is connected as described above. Connections running between the micro chip and RAM are not available at the pins. The DS2250–64 has a second SRAM on $\overline{CE2}$. The timekeeping versions also have the real–time clock connected to $\overline{CE2}$. A block diagram in Figure 5–2 shows the module configuration with 32K RAM and a real–time clock. This is identical for DS2250 or DS5000 modules. These are functionally identical and only differ in form factor.

SECTION 7: POWER MANAGEMENT

Introduction

All Dallas Semiconductor microcontrollers are implemented using fully static CMOS circuitry for low power consumption. Power consumption is a linear function of crystal frequency. Two software initiated modes are available for further power saving at times when processing is not required and V_{CC} is at normal operating voltage. These are the Idle and Stop modes. The additional third mode is the Data Retention or Zero Power State which is made possible by the on–chip, circuitry. The control and status bits which apply to these operating modes are contained in the PCON register and are summarized in Figure 7–1. In addition, Table 7–1 summarizes the state of external pins in each of these modes.

Idle Mode

The Idle mode suspends activity of the CPU. However, the on–chip I/O function, including the timer/counters, and serial port continue their operation. This greatly reduces the number of switching nodes and thereby dramatically reduces the total power consumption of the device. The Idle mode is useful for applications in which lower power consumption is desired with fast response to external interrupts but no other processing.

Software can invoke the Idle mode by setting the IDL bit in the PCON register (PCON.0) to a logic 1 as shown in Figure 7–1. The instruction which sets this bit will be the last instruction executed before Idle mode operation begins. Once in the Idle mode, the microprocessor preserves the entire CPU status including the Stack Pointer, Program Counter, Program Status Word, Accumulator, and RAM. There are two ways to terminate the Idle mode. The first is from an interrupt which has been previously enabled prior to entering Idle mode. This will clear the IDL bit in the PCON register and will cause the CPU to enter the interrupt service routine as normal. When the RETI instruction is executed, the next instruction which will be executed is the one which immediately follows the instruction that set the IDL bit.

The second method of terminating the Idle mode is by a Reset. At this time the IDL bit is cleared and the CPU is placed in the reset state. Since the clock oscillator continues to run in the Idle mode, an oscillator start up delay (referred to as t_{POR} in the AC Electrical Specifications) will not be generated following the reset. Two machine cycles are required to complete the reset operation (24 oscillator periods). It should be noted that the Watchdog Timer continues to run during Idle and that a reset from the on–chip Watchdog Timer will terminate Idle mode.

CONTROL/STATUS BITS FOR POWER CONTROL Figure 7–1

Bit Description:

PCON.6:	**$\overline{POR}$**
"Power On Reset"	Indicates that the previous reset was initiated during a Power On sequence.
Initialization:	Cleared to a 0 when a Power On Reset occurs. Remains at 0 until it is set to a 1 by software.
Read Access:	Can be read normally at any time.
Write Access:	Can be written only by using the Timed Access register.
PCON.5:	**PFW**
"Power Fail Warning"	Indicates that a potential power failure is in progress. Set to a 1 when V_{CC} voltage is below the V_{PFW} threshold. Cleared to a 0 immediately following a read of the PCON register. Once set, it will remain set until read regardless of V_{CC}.
Initialization:	Cleared to a 0 during a Power–On Reset.
Read Access:	Can be read normally at any time.